

U.S. PATENT APPLICATION

Inventor(s): Yoshikazu FUJITA

Invention: LINE DRAWING IMAGE GENERATING DEVICE, STORAGE MEDIUM,
AND LINE DRAWING IMAGE GENERATING METHOD

***NIXON & VANDERHYE P.C.
ATTORNEYS AT LAW
1100 NORTH GLEBE ROAD, 8TH FLOOR
ARLINGTON, VIRGINIA 22201-4714
(703) 816-4000
Facsimile (703) 816-4100***

SPECIFICATION

TITLE OF THE INVENTION

LINE DRAWING IMAGE GENERATING DEVICE, STORAGE MEDIUM, AND LINE DRAWING IMAGE GENERATING METHOD

5 BACKGROUND OF THE INVENTION

Field of the Invention

[0001] The present invention relates to a line drawing image generating device, a storage medium, and a line drawing image generating method. More particularly, the present invention
10 relates to a line drawing image generating device, a storage medium, and a line drawing image generating method for generating line drawing data based on original image data.

Description of the Background Art

15 [0002] Conventionally, there is a technique for extracting outlines of an original image by performing a predetermined process for original image data (for example, see Japanese Patent Gazette No. 2627744). In a typical outline extracting process, a brightness or color difference between a pixel to be processed
20 and a pixel in a neighboring area (for example, 3×3 pixels) of the pixel to be processed is estimated, and the pixel to be processed is extracted as an outline if the brightness or color difference exceeds a predetermined value. The neighboring area of the pixel to be processed is extended (for example, 5×5 pixels), thereby
25 thickening a width of an outline to be extracted and improving

the continuity of the outline.

[0003] By the above outline extracting process, line drawing data is generated from arbitrary image data, and a line drawing is displayed on a TV monitor based on the line drawing data. A
5 user may paint the line drawing using a paint program.

[0004] Now, in an image used in animation (referred to as an animation cel), an outline of a character is typically black (this black outline is referred to as an ink line), and patterns of the character's clothes and grasses in a background are not especially
10 outlined in black (see FIG. 13). If a conventional outline extracting process is applied to the above image in which the outlines are partially thickened, outlines in a resultant line drawing vary widely in line width. However, a neighboring area of a pixel to be processed, which is referred to when a brightness
15 or color difference is estimated, is extended in order to improve the continuity of the outline of the grasses in the background of the image shown in FIG. 13, for example, the outline of the character is unnecessarily thickened in a resultant line drawing as shown in FIG. 14. As a result, the image becomes artificial,
20 which is a problem.

SUMMARY OF THE INVENTION

[0005] Therefore, an object of the present invention is to provide a line drawing image generating device, a program, and
25 a method which generate a natural line drawing from an image such

as an image used in animation, for example, in which outlines are partially thickened, by performing an outline extracting process.

[0006] The present invention has the following features to attain the object mentioned above (notes in parentheses indicate
5 exemplary elements which can be found in the embodiments to follow, though such notes are not intended to limit the scope of the invention).

[0007] The present invention is directed to a line drawing image generating device for generating line drawing data based on
10 original image data (FIG. 13), comprising: ink line area detecting means (a CPU 36 executing step S22); neighboring area detecting means (the CPU 36 executing step S34); outline area detecting means (the CPU 36 executing step S35); line drawing data storing means for storing the line drawing data (84); and color data writing
15 means (the CPU 36 executing steps S31 and S37). The ink line area detecting means detects an area (a pixel for which determination (YES) is made at step S22) whose brightness is smaller than a predetermined value (L) in an original image as an ink line area. The neighboring area detecting means detects a neighboring area
20 (a pixel for which determination (YES) is made at step S34) of the ink line area, which surrounds the ink line area. The outline area detecting means detects an outline portion (a pixel for which determination (YES) is made at step S36) of an image as an outline area, with respect to an area (a pixel for which determination
25 (NO) is made at step S34) other than the ink line area and the

neighboring area in the original image. The color data writing means writes color data (black) to a storage area (a pixel for which determination (YES) is made at step S32 and a pixel for which determination (YES) is made at step S36) of the line drawing data storing means, which corresponds to the ink line area and the outline area, and writes different color data (white) to a storage area of the line drawing data storing means, which corresponds to an area other than the ink line area and the outline area.

[0008] In the case where the original image contains a plurality of pixels (640×480 pixels), the ink line area is included in a predetermined area (3×3 pixels) surrounding a pixel to be processed in the original image, and if the pixel to be processed is not included in the ink line area, the neighboring area detecting means may detect the pixel to be processed as the neighboring area.

15 [0009] Also, the ink line area detecting means may detect, as the ink line area, a portion of an area where brightness is smaller than a predetermined value (L), such that the portion lies near the outline of the area.

[0010] Furthermore, in the case where the original image contains a plurality of pixels (640×480 pixels), and an area other than the ink line area is included in a predetermined area (3×3 pixels) surrounding a pixel to be processed included in an area whose brightness is smaller than a predetermined value (L), the ink line area detecting means may detect the pixel to be processed as the ink line area.

[0011] Still further, the line drawing image generating device may further comprise still image data extracting means (the CPU 36 executing step S11) for extracting arbitrary still image data from moving image data, and use the still image data, which is
5 extracted by the still image data extracting means, as the original image data (FIG. 13) for generating the line drawing data.

[0012] A storage medium of the present invention is directed to a computer readable storage medium (18) storing a line drawing image generating program for generating line drawing data based
10 on original image data (FIG. 13). The line drawing image generating program causes a computer (36) to execute: an ink line area detecting step (S22); a neighboring area detecting step (S34); an outline area detecting step (S35); and a color data writing step (S31, S37). The ink line area detecting step detects an area (a pixel
15 for which determination (YES) is made at step S22) whose brightness is smaller than a predetermined value (L) in an original image as an ink line area. The neighboring area detecting step detects a neighboring area (a pixel for which determination (YES) is made at step S34) of the ink line area, which surrounds the ink line
20 area. The outline area detecting step detects an outline portion (a pixel for which determination (YES) is made at step S36) of an image as an outline area with respect to an area (a pixel for which determination (NO) is made at step S34) other than the ink line area and the neighboring area in the original image. The
25 color data writing step writes color data (black) to a storage

area (a pixel for which determination (YES) is made at step S32 and a pixel for which determination (YES) is made at step S36) of line drawing data storing means (84) for storing the line drawing data, which corresponds to the ink line area and the outline area, and writes different color data (white) to a storage area of the line drawing data storing means (84), which corresponds to an area other than the ink line area and the outline area.

[0013] A line drawing image generating method of the present invention generates line drawing data based on original image data (FIG. 13), comprising: an ink line area detecting step (S22); a neighboring area detecting step (S34); an outline area detecting step (S35); and a color data writing step (S31, S37). The ink line area detecting step detects an area (a pixel for which determination (YES) is made at step S22) whose brightness is smaller than a predetermined value (L) in an original image as an ink line area. The neighboring area detecting step detects a neighboring area (a pixel for which determination (YES) is made at step S34) of the ink line area, which surrounds the ink line area. The outline area detecting step detects an outline portion (a pixel for which determination (YES) is made at step S36) of an image as an outline area with respect to an area (a pixel for which determination (NO) is made at step S34) other than the ink line area and the neighboring area in the original image. The color data writing step writes color data (black) to a storage area (a pixel for which determination (YES) is made at step S32 and a pixel for which determination (YES)

is made at step S36) of line drawing data storing means (84) for storing the line drawing data, which corresponds to the ink line area and the outline area, and writes different color data (white) to a storage area of the line drawing data storing means (84),
5 which corresponds to an area other than the ink line area and the outline area.

[0014] As such, the present invention enables a natural line drawing to be generated from an image such as an image used in animation, for example, in which outlines are partially thickened,
10 by performing an outline extracting process.

[0015] These and other objects, features, aspects and advantages of the present invention will become more apparent from the following detailed description of the present invention when taken in conjunction with the accompanying drawings.

15

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] FIG. 1 is an external view of a game system according to an embodiment of the present invention;

FIG. 2 is a block diagram showing the structure of a
20 main unit;

FIG. 3 is an illustration showing a memory map of a main memory;

FIG. 4 is a flowchart showing an operation of a CPU 36;

FIG. 5 is a portion of a flowchart showing a line drawing
25 generating process;

FIG. 6 is a portion of a flowchart showing a line drawing generating process;

FIG. 7 is an illustration showing exemplary image data to be finally stored in an ink line storing buffer;

5 FIG. 8 is an illustration for describing an outline extracting algorithm;

FIG. 9 is an illustration showing exemplary line drawing data to be finally stored in a line drawing storing buffer;

10 FIG. 10 is a flowchart showing a process of a variant of the embodiment of the present invention;

FIG. 11 is an illustration showing exemplary image data to be finally stored in an additional buffer;

15 FIG. 12 is an illustration showing exemplary line drawing data to be finally stored in the line drawing storing buffer in the variant of the embodiment;

FIG. 13 is an illustration showing an exemplary original image; and

FIG. 14 is an illustration showing a line drawing generated by a conventional outline extracting process.

20

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0017] Hereinafter, a game system according to an embodiment of the present invention will be described.

25 FIG. 1 is an external view of a game system according to the embodiment of the present invention, and FIG. 2 is a block

diagram of the above game system. As shown in FIGS. 1 and 2, a game system 10 includes a main body 12, a DVD-ROM 18, an external memory card 30, a controller 22, a loudspeaker 34a, and a TV monitor 34. The DVD-ROM 18 and the external memory card 30 are removably mounted on and inserted into the main body 12, respectively. The controller 22 is connected to any one of a plurality of controller port connectors (in FIG. 1, four controller port connectors) of the main body 12, via a communication cable. The TV monitor 34 and the loudspeaker 34a are connected to the main body 12 by an AV cable, etc. Note that the main body 12 may perform radio communications with the controller 22. Hereinafter, each component of the game system 10 will be described in details.

[0018] The DVD-ROM 18, which fixedly stores a game program and moving image data, etc., is mounted on the main body 12 when a player plays a game. Note that an arbitrary computer readable storage medium such as a CD-ROM, a MO, a memory card, and a ROM cartridge, etc., may be used in place of the DVD-ROM 18 in order to store the game program, etc.

[0019] The external memory card 30 is a rewritable storage medium such as a flash memory, for example, and stores data such as saving data in the game.

[0020] The main body 12 reads the game program and the moving image data stored in the DVD-ROM 18, and performs a game process.

[0021] The controller 22 is an input device by which the player performs an input about a game operation, and includes a plurality

of operation switches. The controller 22 outputs operation data to the main body 12 when the player presses the operation switches, for example.

[0022] The TV monitor 34 displays the image data output from the main body 12 on a screen. Note that the loudspeaker 34a, which is typically built in the TV monitor 34, outputs a sound of the game output from the main body 12.

[0023] Next, the structure of the main body 12 will be described. In FIG. 2, the main body 12 includes a CPU 36 and a memory controller 38, which is connected to the CPU 36. Furthermore, in the main body 12, the memory controller 38 is connected to a graphics processing unit (GPU) 42, a main memory 40, a DSP 44, and various interfaces (I/F) 48 to 56. The memory controller 38 controls data transfer between the above components.

[0024] When the game is started, a disk drive 16 drives the DVD-ROM 18 mounted on the main body 12. The game program stored in the DVD-ROM 18 is read by the main memory 40 via the disk I/F 56 and the memory controller 38. The CPU 36 executes the program in the main memory 40, and starts the game. After the game is started, the player uses the operation switches for inputting game operations, etc., to the controller 22. In accordance with the input from the player, the controller 22 outputs operation data to the main body 12. The operation data output from the controller 22 is input to the CPU 36 via the controller I/F 48 and the memory controller 38. The CPU 36 performs a game process in accordance

with the input operation data. The GPU 42 and the DSP 44 are used for generating image data, etc., in the game process. An ARAM (Audio-grade DRAM) 46 is used when the DSP 44 performs a predetermined process.

5 [0025] The GPU 42 generates image data to be displayed on the TV monitor 34, and outputs the image data to the TV monitor 34 via the memory controller 38 and the video I/F 50. Note that sound data generated by the CPU 36 when the game program is executed is output from the memory controller 38 to the loudspeaker 34a
10 via the audio I/F 54.

[0026] In FIG. 3, an illustration showing a memory map of the main memory 40 is shown. The main memory 40 includes a program storage area 58 for storing a program, and a data storage area 76 for storing data.

15 [0027] The program storage area 58 stores a game main processing program 60, a moving image reproducing program 62, a still image capturing program 64, a line drawing generating program 66, and a paint program 74, etc. These programs are read by the DVD-ROM 18, if necessary, and stored in the main memory 40. The line drawing
20 generating program 66 includes an ink line detecting program 68, an ink line neighboring area detecting program 70, and an outline extracting program 72.

[0028] The data storage area 76 includes a moving image data storage area 78, a display buffer 80, an ink line image buffer
25 82, and a line drawing storing buffer 84, and further stores a

paint program image data 86, etc. The moving image data storage area 78 temporarily stores the moving image data read from a DVD-ROM. The paint program image data 86 is read from the DVD-ROM 18, if necessary, and stored in the main memory 40.

5 [0029] Hereinafter, an operation of the game system 10 will be described.

In the present embodiment, the player selects a moving image when the game is started, and a reproducing process is performed for the selected moving image. The player is allowed
10 to extract a currently displayed image as a still image by operating the controller 22 at any given time during reproduction of the moving images. The still image is extracted, and a line drawing is generated from the extracted still image. Then, the player enjoys freely painting the generated line drawing on the TV monitor
15 34.

[0030] With reference to flowcharts shown in FIGS. 4 to 6, a process performed by the CPU 36 of the present embodiment will be described.

In FIG. 4, when the game process is started, the CPU
20 36 performs a moving image reproducing process based on the moving image reproducing program 62. Specifically, the CPU 36 transfers image data of a predetermined frame, which is included in the moving image data stored in the DVD-ROM 18 (or temporarily stored in the moving image data storage area 78 after being read from the DVD-ROM
25 18) to the display buffer 80 (S11), and outputs a video signal

based on the image data, which is transferred to the display buffer 80, to the TV monitor 34 (S13). Then, the CPU 36 determines whether or not the moving image reproducing process is ended (S14), and performs the same process for image data of a next frame if the moving image reproducing process is not ended. By repeating the process performed at step S11 and step S13, moving images are reproduced on the screen of the TV monitor 34. If the determination is made at step S13 that the moving image reproducing process is ended, the game process is ended.

10 [0031] Note that, during the moving image reproducing process, the CPU 36 monitors the input from the controller 22, based on the still image capturing program 64 (S12), and stops the moving image reproducing process if there is an input from the controller 22. Then, the CPU 36 uses the still image data currently stored in the display buffer 80 as original image data, and starts a line drawing generating process based on the line drawing generating program 66 (S15). The details of the line drawing generating process will be described below.

20 [0032] As a result of the line drawing generating process, line drawing data is stored in the line drawing storing buffer 84. The CPU 36 transfers the line drawing data to the display buffer 80 (S16), and outputs a video signal based on the line drawing data, which is transferred to the display buffer 80, to the TV monitor 34 (S17). The CPU 36 executes a line drawing painting process by the paint program utilizing the line drawing data (S18), and

25

ends the game.

[0033] Next, with reference to FIGS. 5 and 6, the details of the line drawing generating process at step S15 will be described.

In FIG. 5, when the line drawing generating process is started, a process for detecting an ink line area (an area with a low brightness, such as an ink line of an animation cel) in an original image is performed first based on the ink line detecting program 68. Specifically, the CPU 36 clears the ink line image buffer 82 (writes white to all pixels), and sets 1 to X and Y (coordinates for specifying a pixel to be processed), respectively, as an initial value, and sets 10 as a threshold value L of a brightness used for determination of an ink line area (S21). Note that, in this embodiment, it is assumed that a brightness of each pixel of image data is defined within a range from 0 to 255. The threshold value L, which is not limited to 10, is set to an optimum value.

[0034] The CPU 36 determines whether or not a brightness of the pixel (X, Y) to be processed in the original image is equal to or smaller than L, based on the original image data stored in the display buffer 80 (S22). If the determination is made that the brightness is equal to or smaller than L, the CPU 36 writes black to a storage area corresponding to a pixel (X, Y) of the ink line image buffer 82 (S23), and proceeds to step S24. On the other hand, if the brightness is greater than L, the CPU 36 directly proceeds to step S24.

[0035] The CPU 36 increments X at step S24, and determines at

step S25 whether or not X is smaller than 640. Note that, in this embodiment, it is assumed that the size of image data to be processed is 640×480 pixels. If the determination is made that X is equal to or smaller than 640, the CPU 36 goes back to step S22. If the
5 determination is made that X is greater than 640, the CPU 36 proceeds to step S26. At step S26, the CPU 36 determines whether or not Y is 480. If the determination is made that Y is not 480, the CPU 36 increments Y at step S27, sets 1 to X, and goes back to step S22. On the other hand, if Y is 480, the CPU 36 proceeds
10 to step S31 of FIG. 6. By the process performed at steps S24 to S27, the pixel to be processed is sequentially shifted, which is just like scanning a scanning line, and all pixels are finally processed.

[0036] As a result of the above process from steps S21 to S27,
15 in the ink line image buffer 82, black is written to the pixels corresponding to the ink line area (area with a low brightness) of the original image, and white is written to the remaining pixels, as shown in FIG. 7.

[0037] In FIG. 6, the CPU 36 first clears the line drawing storing
20 buffer 84 (writes white to all pixels), and sets 1 to X and Y (coordinates for specifying a pixel to be processed), respectively, as an initial value (S31). Then, the CPU 36 determines whether or not the pixel (X, Y) to be processed is black in the ink line image buffer 82 (that is, whether or not the pixel to be processed
25 is included in the ink line area) (S32). If the determination

is made that the pixel to be processed is black, the CPU 36 writes black to a pixel (X, Y) of the line drawing storing buffer 84 at step S37, and proceeds to step S38. On the other hand, if the pixel to be processed is not black, the CPU 36 proceeds to step
5 S33.

[0038] The CPU 36 determines at step S33 whether or not X is between or equal to 2 and 639, and Y is between or equal to 2 and 479. If the determination is made that X is between or equal to 2 and 639, and Y is between or equal to 2 and 479, the CPU 36 proceeds
10 to step S34. On the other hand, if the determination is made that X is not between or equal to 2 and 639, and Y is not between or equal to 2 and 479, the CPU 36 proceeds to step S38. As a result, if the pixel to be processed is included in the pixels on the edge (upper or bottom edge, right or left edge) of the image data, a
15 process to be performed in the following steps S34 to S36 is omitted.

[0039] As step S34, the CPU 36 determines whether or not any of neighboring pixels of the pixel (X, Y) to be processed is black in the ink line image buffer 82 (that is, whether or not the pixel to be processed is in a vicinity of the ink line area), based on
20 the ink line neighboring area detecting program 70. Note that the neighboring pixels of the pixel (X, Y) to be processed are included in a predetermined area (in this embodiment, 3×3 pixels) surrounding the pixel to be processed. More specifically, the neighboring pixels of the pixel (X, Y) to be processed are the
25 following eight pixels: (X-1, Y-1), (X, Y-1), (X+1, Y-1), (X-1,

Y), (X+1, Y), (X-1, Y+1), (X, Y+1), (X+1, Y+1). If the determination is made that any of neighboring pixels of the pixel (X, Y) to be processed is black, the CPU 36 proceeds to step S38. On the other hand, if no neighboring pixels of the pixel (X, Y) to be processed are black, the CPU 36 proceeds to step S35. That is, an outline extracting process of step S35, which will be described below, is applied only to a pixel (X, Y) which is not included in the ink line area and its neighboring area.

[0040] At step S35, the CPU 36 performs the outline extracting process for the pixel (X, Y) to be processed, which is a pixel of the display buffer 80, based on the outline extracting program 72. As the above outline extracting process, it is possible to use an arbitrary well-known outline extracting algorithm. In this embodiment, an outline extracting process using a Sobel operator, which is a type of differential operator, is performed as an example.

[0041] As shown in FIG. 8 (a), the pixels in a predetermined area (3×3 pixels) surrounding the pixel to be processed are referred to as pixels A to I, respectively. Brightness values of the pixels A to I are referred to as L (A) to L (I), respectively. In the outline extracting process using the Sobel operator, these brightness values of the nine pixels are weighted by differential operators as shown in FIG. 8 (b1) and FIG. 8 (b2), and the weighted sum is obtained. Thus, a brightness value variation in a horizontal direction and a brightness value variation in a vertical direction are obtained as shown in FIGS. 8 (c1) and 8 (c2), respectively. A

brightness variation vector is determined based on the above brightness value variations. That is, a horizontal component of the brightness variation vector corresponds to the brightness value variation in a horizontal direction, and a vertical component of the brightness variation vector corresponds to the brightness value variation in a vertical direction. An outline is a portion in which a brightness value fluctuates widely, whereby it is possible to extract the outline by extracting a portion whose magnitude of a brightness variation vector is greater than a predetermined threshold value. Note that it is possible to differently change the results of the outline extracting process, for example, by extending the range of applicability of the differential operator, or adjusting a threshold value used for estimating a magnitude of a brightness variation vector. For example, it is possible to improve the continuity of an outline to be extracted by reducing the threshold value used for estimating a magnitude of a brightness variation vector. However, if the threshold value used for estimating a magnitude of a brightness variation vector is reduced in a conventional outline extracting process, ink line portions are unnecessarily thickened as shown in FIG. 14.

[0042] At step S36, the CPU 36 determines whether or not the pixel (X, Y) to be processed is included in the outline portion, based on the processing results at step S35. If the determination is made that the pixel (X, Y) to be processed is included in the outline portion, the CPU 36 writes black to a pixel (X, Y) of the

line drawing storing buffer 84 at step S37, and proceeds to step S38. On the other hand, if the pixel (X, Y) to be processed is not included in the outline portion, the CPU 36 directly proceeds to step S38.

5 **[0043]** The process performed at steps S38 to S41 is similar to the process performed at steps S24 to S27 shown in FIG. 5. The above process is performed for sequentially shifting the pixel to be processed. After the process performed at steps S38 to S41, all pixels are processed, and the line drawing generating process
10 is ended.

[0044] As a result of the above line drawing generating process, in the line drawing storing buffer 84, black is written to the pixels corresponding to the ink line area of the original image, and the pixels corresponding to the outline portion in an area
15 other than the above ink line area and its neighboring area, and white is written to the remaining pixels as shown in FIG. 9. The line drawing data stored in the line drawing storing buffer 84 is used for a line drawing to which the player applies his/her desired colors. As such, according to the present embodiment,
20 the outline extracting process is not performed for the pixels in the vicinity of the ink line area, thereby preventing the ink line portion in the line drawing from being thickened compared to the original image. Thus, in the case where the results of the outline extracting process at step S35 are changed in order
25 to improve the continuity of the outlines of the grasses in the

background and the outlines of the patterns of the character's clothes, the ink line portion is not thickened accordingly. As a result, as shown in FIG. 9, it is possible to obtain a natural line drawing whose outline continues smoothly.

5 [0045] Note that, the line drawing shown in FIG. 9 includes black areas other than the outline, for example, the character's pupils and shadows on the forehead, and the player may desire to freely paint these areas. Thus, hereinafter, a method for generating a line drawing which enables the player to freely paint
10 these areas will be described.

[0046] In this case, a new buffer (hereinafter, referred to as an additional buffer) is added to the data storage area 76 of the main memory 40, and a process shown in FIG. 10 is added between step S26 shown in FIG. 5 and step S31 shown in FIG. 6. That is,
15 the CPU 36 first copies the data of the ink line image buffer 82 to the additional buffer, and sets 1 to X and Y (coordinates for specifying a pixel to be processed), respectively, as an initial value (S51). Then, the CPU 36 determines whether or not all pixels in a neighboring area (in this case, an area of 5×5 pixels surrounding
20 the pixel (X, Y) to be processed) are black in the ink line image buffer 82 (S52). If the determination is made that all pixels in the neighboring area are black, the CPU 36 writes white to a pixel (X, Y) of the additional buffer at step S53, and proceeds to step S54. On the other hand, if all pixels in the neighboring
25 area are not black, the CPU 36 directly proceeds to step S54. As

a result, white is written to a pixel in the ink line area, such that the pixel lies away from the outline.

[0047] The process from steps S54 to S57 is similar to the process from steps S24 to S27 of FIG. 5. The above process is performed
5 for sequentially shifting the pixel to be processed. When all pixels are processed, the CPU 36 proceeds to step S31 of FIG. 6. In the additional buffer, image data shown in FIG. 11 is stored. In the process after step S31, the additional buffer is referred to in place of the ink line image buffer 82. As a result of the
10 above line drawing generating process, line drawing data shown in FIG. 12 is stored in the line drawing storing buffer 84. The player can enjoy freely painting the character's pupils and shadows on the forehead, for example, based on the line drawing data.

[0048] Note that, in the present embodiment, a brightness value
15 of the pixel to be processed is used for determining whether or not the pixel to be processed is in the ink line area, but the present inventions are not limited thereto. For example, values of R, G, and B of the pixel to be processed may be used for performing the above determination.

[0049] Also, in the present embodiment, a predetermined process
20 is performed by referring to an area of 3×3 pixels surrounding the pixel to be processed at steps S34 and S35, but the present invention is not limited thereto. For example, an area of 5×5 pixels may be referred to. Also at step S52, a neighboring area
25 is not limited to an area of 5×5 pixels.

[0050] Furthermore, in the present embodiment, it is assumed that either white or black is written to each pixel of the line drawing storing buffer, but the present invention is not limited thereto. For example, red, blue, or brown may be written to each
5 pixel of the line drawing storing buffer.

[0051] While the invention has been described in detail, the foregoing description is in all aspects illustrative and not restrictive. It is understood that numerous other modifications and variations can be devised without departing from the scope
10 of the invention.